Interspeech 2018
2-6 September 2018, Hyderabad

Neural network architecture that combines temporal and summative features for infant cry classification in the Interspeech 2018 Computational Paralinguistics Challenge

Mark Huckvale

Speech, Hearing and Phonetic Sciences, University College London, U.K.

m.huckvale@ucl.ac.uk

Abstract

This paper describes the application of a novel deep neural network architecture to the classification of infant vocalisations as part of the Interspeech 2018 Computational Paralinguistics Challenge. Previous approaches to infant cry classification have either applied a statistical classifier to summative features of the whole cry, or applied a syntactic pattern recognition technique to a temporal sequence of features. In this work we explore a deep neural network architecture that exploits both temporal and summative features to make a joint classification. The temporal input comprises centi-second frames of low-level signal features which are input to LSTM nodes, while the summative vector comprises a large set of statistical functionals of the same frames that are input to MLP nodes. The combined network is jointly optimized and evaluated using leave-one-speaker-out cross-validation on the challenge training set. Results are compared to independently-trained temporal and summative networks and to a baseline SVM classifier. The combined model outperforms the other models and the challenge baseline on the training set. While problems remain in finding the best configuration and training protocol for such networks, the approach seems promising for future signal classification tasks.

Index Terms: computational paralinguistics, infant cry, deep neural networks, time series

1. Introduction

1.1. Task

The goal of the Interspeech 2018 Cry Challenge was to classify vocalisations of infants from short audio recordings. The training and testing corpus (CRIED) was provided by the Department of Phoniatrics, Medical University of Graz [1]. It consists of 5587 vocalisations made by 20 infants while lying unattended in their cots. First recordings were made when the infants were 4 weeks old, and the last when they were 16 weeks. Selected excerpts of the recordings were classified by two experts in the field of early language development into three classes: (i) neutral or positive mood sounds, (ii) fussing sounds, and (iii) crying sounds. For further details of the corpus and the challenge, please see [2].

In this paper, we develop and evaluate some neural-network architectures for infant cry classification. In this we have built on our previous investigations into the classification of other paralinguistic properties of the voice: for Cognitive Load [3], for Fatigue [4], and for the Common Cold [5]. The strategy in these previous studies has been to create well-motivated feature sets that capture temporal, spectral and modulational properties of each audio token relevant for the task, and then summarize those features over each token using a number of statistical functions. These summative features then describe each token in terms of a fixed-length vector which may be used to train support-vector machine (SVM) or deep-neural network (DNN) classifiers.

A disadvantage of the use of summative feature vectors for classification is that the choice of summarizing functions is made *a priori*, before any analysis of the classification problem, and these may be irrelevant, redundant or less than optimal for the task. The summarizing functions also make assumptions about the distribution of useful information within the temporal sequence, for example that all frames of data are equally important.

An alternative to summarising the time series would be to use a statistical pattern recognition technique on the temporal sequence itself. However the token labels describe the whole sequence rather than its parts, so the problem then is to create appropriate training labels for each part of the sequence. An approach presented in the Interspeech 2017 challenge [6] was to divide the time sequence into overlapping fixed length sections, and build a classifier to label each section with the sequence label. The resulting time sequence of class probabilities was then summed and input to an SVM classifier to make an overall classification.

In this article we develop this temporal sequence classification approach further and apply it to infant cry classification. We replace the fixed length temporal feature windows used in [6] with a bidirectional LSTM (long short-term memory) network over the whole sequence. We evaluate the temporal sequence classifier against one trained on summative features. Finally we construct a neural network architecture that inputs both temporal and summative features to make best use of both.

Section 2 of this paper looks at the typical acoustic characteristics of infant cry, and how the problem of infant cry recognition has been approached in previous studies. Section 3 describes the methods by which features are extracted and classifiers are trained and evaluated, while section 4 presents the performance of the classifiers on the challenge corpus and discusses the outcomes with respect to baseline scores. The paper concludes with a discussion of the promise for the new approach to temporal sequence classification.

10.21437/Interspeech.2018-1959

2. Infant Cry Classification

The automatic classification of infant vocalisations has a long history, going back over 20 years; see [7] for a review of supervised machine learning approaches. Some studies have simply tried to classify vocalisations into basic types such as hunger, discomfort and pain, while others have sought to make early diagnosis of significant medical conditions such as brain haemorrhage, asphyxia, deafness or Down syndrome.

The different studies vary considerably in many ways: the nature of the cry corpus and labels, the choice of acoustic features, and the choice of classifier. This makes it hard to compare the performance of different systems. However some common patterns can be seen. Many systems are based on a summative feature vector generated by applying statistical functions to a time-series of acoustic features [7]. Acoustic features usually include spectral envelope information, sometimes together with information about fundamental frequency, voicing, and voice quality. A few studies have exploited syntactic pattern recognition systems, such as HMM [8] or GMM [9], based on MFCC and F0 time-series.

Previous studies that have simply tried to characterise infant cry rather than build a classifier (e.g. [10]) often describe a richer set of acoustic features. For example they describe different cries using features related to the temporal development of the vocalisation, such as the duration and intervals between cries, the occurrence of discontinuities in pitch or in voicing, or of the timing of vocalisations with respect to breathing. These studies suggest that there may be additional information about the nature of the cry to be found in the temporal pattern than has been used so far for automatic classification.

As a preliminary to our study we explored the essential acoustic properties of the cries in the CRIED corpus as a function of labelled category. Fig.1 shows variation in duration, pitch height, pitch range, pitch perturbation, energy and voicing for the neutral, fussing and crying labels over all tokens for all speakers in the training set.

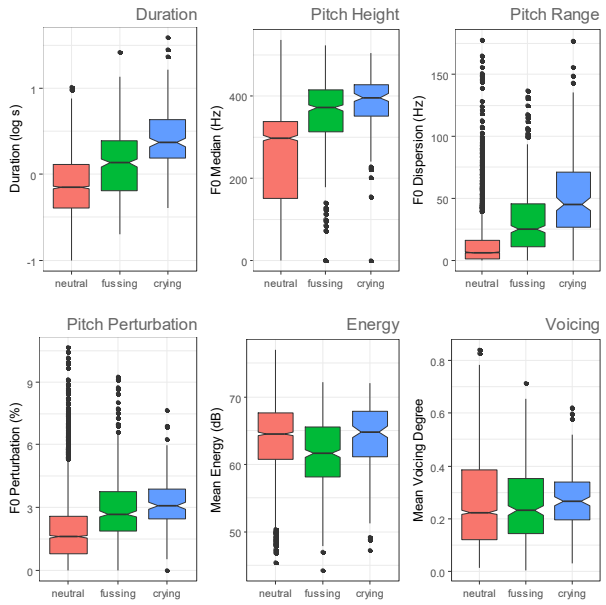

Figure 1. Variation in basic acoustic parameters with infant vocalisation class

Duration was calculated from an automatic end-pointing based on energy, pitch height was calculated from median F0, pitch range was median absolute dispersion of F0, pitch perturbation was mean absolute % change in F0 across frames, energy was calculated from the amplitude envelope smoothed with 50ms hamming window, voicing degree was a composite measure of periodicity based on measures of energy, autocorrelation and zero-crossing rate. All measures were calculated with the SFS toolkit [11].

There are clear differences across categories for these simple summative features. Fitting a linear mixed effects model with speaker as a random factor shows significant differences ($p<0.05$) in mean duration, pitch height, pitch range and pitch perturbation across all three categories, while energy is only different for the fussing category. Voicing degree is not significantly different across categories. These six features alone have moderate success in discriminating categories of cry. A CART model on these acoustic parameters achieves an unweighted accuracy of 60.7% on the training set using leave-one-speaker-out cross-validation.

Using these acoustic measures as guide, we are now able to identify typical vocalisations of each class, see Fig 2. What is noticeable is that there are differences in the temporal development of the cries as well as in their spectral properties. This suggests that an approach that is sensitive to temporal patterning might provide additional useful information for recognition.

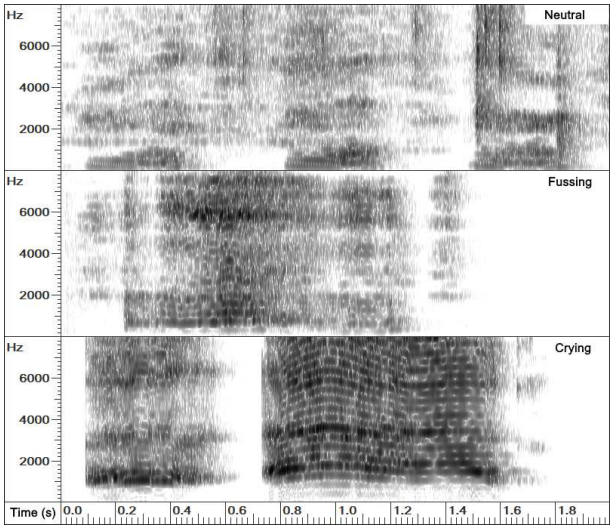

Figure 2. Spectrograms of typical Neutral, Fussing and Crying vocalisations.

3. Methods

3.1. Feature extraction and normalisation

The acoustic features used in the experiments were generated using the OpenSMILE toolkit [12] and the ComParE_2016 configuration as used in the challenge baseline [2]. The temporal feature set comprised the low-level-descriptor (LLD) frames generated by the toolkit for each recording. The LLD frames contain 126 parameters every 10ms, and describe information about pitch, voicing, voice quality and spectral features along with their temporal derivatives. The summative feature set is the result of applying a large number of statistical functions to the LLDs of a recording, such as measures of

central tendency, range, maximum, minimum, etc. The summative feature set contains 6373 parameters.

Two-types of feature normalisation were investigated: global z-score normalisation is performed across all speakers (i.e. speaker independent), while personal z-score normalisation is applied separately to each speaker (speaker dependent). For global normalisation within leave-one-speaker-out cross validation, the means and standard deviations are calculated using the training speakers only, and then used to normalise the left-out speaker.

3.2. Feature selection

To investigate whether recognition on the basis of summative features is improved by feature selection, the utility of each feature for classification is assessed using the F-statistic. The F-ratio for each feature was used to rank the utility of that feature for each individual speaker. The average rank of each feature over all speakers in the training set was then used to obtain a final best feature ranking. The idea was to find features which had stable high ranks over different speakers. For leave-one-speaker-out cross-validation, the feature ranks were computed using only the training speakers and the best ranked features were applied to the left-out speaker.

3.3. SVM classification

A Support Vector Machine (SVM) was used as a baseline classifier to compare against the neural-network approaches. We use the e1071 package [13] as implemented in the R statistics system [14]. For all experiments, a linear kernel was used and the cost parameter was set to 10^{-5} – the best value found in the baseline.

3.4. Neural networks

The neural networks were built using the Keras toolkit [15] operating with the TensorFlow back-end [16]. All networks had softmax output layers and were trained using RMSProp [17] with a categorical cross-entropy loss function. The same training regime was used for all networks. Sample weighting was used to compensate for class imbalance in the training set. Dropout layers with 25% dropout were used to improve generalisation. Three networks were constructed and trained: one operating on the summative features set, one operating on the temporal feature set, and one operating on both feature sets jointly.

The Summative classifier consisted of two layers of densely connected nodes with tanh activations. Output is the 3-way class probability. The layers were 6373 (input) -64-64-3 (output).

The Temporal classifier consisted of two layers of bidirectional LSTM nodes, feeding a time distributed dense layer. Input temporal sequences of LLD frames were either padded or truncated into 500 frames (i.e. 5s). The LLD frames were right aligned in the input window, and any padding consisted of frames with all zero values. Output labelling was the same 3-way class label applied to every frame including padding frames. The layers were 126x500(input)-2x32x500-2x32x500-3x500(output). For final classification, the outputs were averaged over the 500 time steps to derive class probability scores per token.

The Combined classifier is a combination of the Temporal and Summative classifiers within one neural network model. The two classifiers are joined at the output of their second layers: the output of the second dense layer in the summative classifier and the averaged output of the second LSTM layer in the temporal classifier, see Fig.3. The network has two sets of outputs: the main output is a three-way softmax classification of class probabilities based on the concatenation of the 64-way outputs of the two second layers. The auxiliary output is the temporally distributed class labels as used to train the temporal classifier. Both outputs are used during training, this ensures that the temporal classifier generates appropriate representations for the classification task within the second layer LSTM. However the training weight associated with the auxiliary output is set to be only 0.25 of the weight given to the main output.

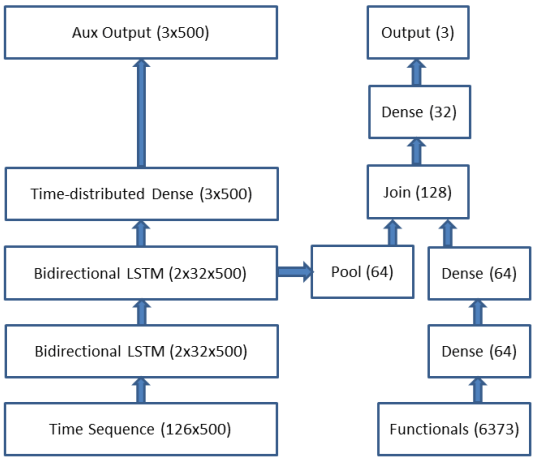

Figure 3. Combined temporal and summative neural-network architecture

3.5. Performance measure

The challenge performance measure is specified as unweighted-average recall (UAR). This is the average of the accuracies of a classifier on each class taken separately. This measure is chosen because the corpus is imbalanced for the different cry labels. The training partition has 2292 neutral, 368 fussing and only 178 crying tokens.

The class imbalance makes the UAR measure rather sensitive to small changes in classifier performance on tokens in the minority classes. Large fluctuations in UAR occur across different classifier configurations even when overall accuracy is relatively stable. This sensitivity makes it hard to find the optimum hyper-parameters for the classifier configurations.

To aid optimisation of the classifiers, a measure UAR_{max} was introduced. This figure represents the best obtainable UAR for a given set of class scores generated by a classifier. To obtain UAR_{max}, the probability of class j for token i is transformed using weights a_j and b_j as:

$$p'_{ij} = \left(a_j p_{ij}\right)^{b_j}$$

where the weights $\{a_j\}$ and $\{b_j\}$ are found by functional optimization across all tokens to maximise UAR. This is effectively the "calibration" step of the multi-class FOCAL toolkit [18]. While UAR_{max} is not necessarily a good measure of how a classifier will perform on unseen data, it does provide a better means to compare classifiers on the same data when searching for their best training hyper-parameters.

4. Results

4.1. Effect of Normalisation and feature selection

We first use the SVM classifier to determine the best form of normalisation and the best number of features suited to this classification task. Fig 4 summarises the effect of feature selection and normalisation on UAR and UAR_{max} estimated using leave-one-speaker-out cross-validation on the training set. For feature selection, best ranked features of size 100, 200, 500, 1000, 2000, 5000 and 6373 were tested. In all cases more features gave higher performance, that is, there seems no benefit in performing feature selection on these data. For normalisation, global and personal z-score normalisation approaches were compared. Global normalisation gave higher performance. This may have been because some speakers did not have any cry vocalisations, so that per-speaker normalisation did not appropriately describe the range of features used by the speaker. Lastly, UAR_{max} is seen to be less sensitive to configuration change than UAR, confirming its utility in comparing classifiers. The fact that the SVM decision shows on occasions a UAR slightly greater than UAR_{max}, may be due to the approximations within the SVM that are used to generate class probabilities.

As a consequence of these findings, global normalisation without feature selection were used for the main investigation.

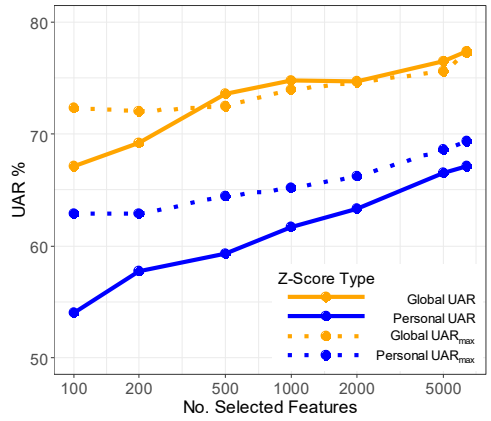

Figure 4. UAR and UAR_{max} on training set for SVM with varying number of selected features and global versus personal z-score normalisation.

4.2. Neural network classification performance

The performance of the neural-network models compared to the SVM baseline on training and test partitions of the CRIED corpus are shown in Table 1. For the training set these are UAR_{max} figures calculated with leave-one-speaker-out cross-validation. For the test set, these are UAR figures for class scores that were transformed using the best weightings found when calculating UAR_{max} on the training set.

The summative classifier, operating on the same features as the SVM gave similar performance on the training set and better performance on the test set, showing that the DNN approach works well. The temporal classifier operating on the LLD frames gave the same test set performance but was slightly worse on the training set, showing that useful information can be extracted from the LLD sequence by this architecture. The combined summative and temporal classifier performed best of all on both corpus partitions, showing that temporal processing was able to extract useful information not present in the openSMILE functionals. Overall performance of the models on the training set matched or exceeded the simple classifiers used in the challenge baseline. However test set performance is worse than the best challenge baseline system (UAR=73.2%). However it should be noted that all neural network scores are sensitive to the configuration and training hyper-parameters, which might still be sub-optimal.

Table 1. Performance of different classifiers on the challenge corpus. Training score is UAR_{max} obtained from leave-one-speaker-out cross-validation.

Classifier	Features	Train UAR_{max}%	Test UAR%
SVM	6373	77.15	66.27
Summative NN	6373	77.34	68.28
Temporal NN	126x500	76.27	68.28
Combined NN	126x500+6373	79.26	68.72

5. Discussion

In this paper we have presented three neural-network architectures for infant cry classification: one based on temporal features, one based on summative features, and one based on both simultaneously. We have shown that all give performance scores that are similar to or exceed the challenge baseline on the training data. Although differences are small, there are encouraging signs that the combined model benefits from having access to both temporal and summative features.

Two areas are still in need of improvement: (i) the necessity of using fixed length temporal sequences in DNN training is a limitation of the Keras toolkit, and could be removed with further algorithm development; (ii) it is still difficult to find the optimal configurations and training protocol for the DNN classifiers. Although we have made steps to improve the evaluation of classifiers through leave-one-speaker-out cross-validation and through the introduction of UAR_{max}, there are likely other strategies that will ensure that test set performance more closely matches that obtained on the training set, particularly in the case of highly imbalanced classes.

In this experiment the summative vector was computed from the same LLD frames as were presented to the temporal classifier. This might seem to be redundant, since surely the temporal classifier has the ability to recreate any statistical functional used to generate the summative vector. However the summative vector allows the researcher to add prior knowledge about useful global characteristics of the temporal sequence, without simply hoping that these will be rediscovered by the model. Future work might determine which statistical functionals add most value to the temporal feature analysis.

6. Acknowledgements

Thanks to the organisers of the Interspeech 2018 Computational Paralinguistics Challenge for making this study possible.

7. References

[1] P. B. Marschik, F. B. Pokorny, R. Peharz, D. Zhang, J. O'Muircheartaigh, H. Roeyers, S. Bölte, A. J. Spittle, B. Urlesberger, B. Schuller, L. Poustka, S. Ozonoff, F. Pernkopf, T. Pock, K. Tammimies, C. Enzinger, M. Krieber, I. Tomantschger, K. D. Bartl-Pokorny, J. Sigafoos, L. Roche, G. Esposito, M. Gugatschka, K. Nielsen-Saines, C. Einspieler, W. E. Kaufmann, The BEE-PRI Study Group: "A novel way to measure and predict development: A heuristic approach to facilitate the early detection of neurodevelopmental disorders*". Current Neurology and Neuroscience Reports*, 17:43, 2017.

[2] B. Schuller, S. Steidl, A. Batliner, P. Marschik, H. Baumeister, F. Dong, F. B. Pokorny, E.-M. Rathner, K. D. Bartl-Pokorny, C. Einspieler, D. Zhang, A. Baird, S. Amiriparian, K. Qian, Z. Ren, M. Schmitt, P. Tzirakis, S. Zafeiriou: "The INTERSPEECH 2018 Computational Paralinguistics Challenge: Atypical & Self-Assessed Affect, Crying & Heart Beats", *Proc. Interspeech 2018*, Hyderabad, 2018.

[3] M. Huckvale, "Prediction of Cognitive Load from Speech with the VOQAL Voice Quality Toolbox for the InterSpeech 2014 Computational Paralinguistics Challenge", *Proc. Interspeech 2014*, Singapore, 2014.

[4] K. Baykaner, M. Huckvale, I. Whiteley, S. Andreeva, O. Ryumin, "Predicting Fatigue and Psychophysiological Test Performance from Speech for Safety-Critical Environments". *Frontiers in Bioengineering and Biotechnology*, 3, 2015.

[5] M. Huckvale, A. Beke, "It sounds like you have a cold! Testing voice features for the Interspeech 2017 Computational Paralinguistics Cold Challenge", *Proc. Interspeech 2017*, Stockholm, 2017.

[6] G. Gosztolya, R. Busa-Fekete, T. Grosz, L. Toth, "DNN-based Feature Extraction and Classifier Combination for Child-Directed Speech, Cold and Snoring Identification", *Proc. Interspeech 2017*, Stockholm, 2017.

[7] T. Fuhr, H. Reetz, C. Wegener, "Comparison of Supervised-learning Models for Infant Cry Classification", *Int. J. Health Professions* 2 (2015) p4-15.

[8] K. Honda, K. Kitahara, S. Matsunaga, M. Yasmashita, K. Shinohara, "Emotion classification of infant cries with consideration for local and global features", *Proc. 2012 Asia Pacific Signal and Information Processing Conference*, 2012.

[9] A. Kumar Singh, J. Mukhopadhyay, K. Sreenivasa Rao, "Classification of Infant Cries Using Epoch and Spectral Features", *IEEE 2013 National Conference on Communications (NCC)* , 2013.

[10] Y. Kheddache, C. Tadj, "Frequential Characterization of Healthy and Pathologic Newborns Cries", *American Journal of Biomedical Engineering* 2013, 3(6): 182-193.

[11] M. Huckvale, "Speech Filing System", 2013, http://www.phon.ucl.ac.uk/resource/sfs.

[12] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE – The Munich Versatile and Fast Open-Source Audio Feature Extractor," in *Proc. of ACM Multimedia*, Florence, Italy, 2010, pp. 1459–1462.

[13] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, A., Leisch, "e1071: Misc Functions of the Department of Statistics TU Wien", 2014, http://CRAN.Rproject.org/package=e1071.

[14] "R: A language and environment for statistical computing". R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/.

[15] F. Chollet and others, "Keras: The Python Deep Learning library", 2018, https://github.com/keras-team/keras.

[16] Martín Abadi, and others, "TensorFlow: Large-scale machine learning on heterogeneous systems", 2015, https://tensorflow.org.

[17] G. Hinton, N. Srivastava, K. Swersky, "Neural Networks for Machine Learning Lecture 6a: Overview of mini-batch gradient descent", 2014, http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[18] N. Brümmer, "FoCal Multi-class Toolkit", 2007, https://sites.google.com/site/nikobrummer/focalmulticlass